Dauphine | PSL
UNIVERSITÉ PARIS

Multi-Level Monte Carlo

Master MASEF

Université Paris-Dauphine

Paul-Antoine Charbit Jérémie Vilpellet

Project for the course
Monte Carlo and Finite Difference Methods

January 2025

Contents

1 Introduction **2**

2 Classical Monte Carlo **2**
2.1 Principle of the method 2
2.2 Euler scheme 3
2.3 Error Decomposition 3

3 Multi Level approach **4**
3.1 Telescoping Sum Formulation 5
3.2 Variance and Cost Reduction 5
3.3 Weak Error, Bias, and Computational Complexity 5
3.4 Bias estimation and stopping criterion 6
3.5 MLMC Algorithm 7

4 Finite difference approach **7**
4.1 Finite Difference Methods for the Black–Scholes PDE 7
4.2 Convergence and Stability of the θ-Scheme for the Black–Scholes PDE 9

5 Applications **9**
5.1 Option Pay-off 9
5.2 Multi Level numerical result 10
5.3 Finite difference numerical result 14

6 Conclusion **15**

1 Introduction

Monte Carlo methods are widely used in finance to approximate the value of financial products, particularly options. Within the framework of the Black-Scholes paradigm (and its extensions), options are valued as the expectation of their actualized pay-off under a specific probability space.

The value of an option depends on the functional of the path of one or several underlying financial instruments, such as, but not limited to, stock prices, exchange rates, or commodity prices. The dynamics of these underlying instruments are typically modeled using stochastic differential equations (SDEs), which often do not admit closed-form solutions.

In this context, numerical methods, and in particular Monte Carlo simulations, play a crucial role. The classical Monte Carlo method approximates the expectation of the payoff by simulating numerous paths of the underlying variables, which are discretized over time. While efficient and robust, this approach can be computationally expensive when high accuracy is required, as the error decreases slowly with the number of simulations.

In this project, we first present the classical Monte Carlo method and discuss its principles and limitations. We then introduce the Multi-Level Monte Carlo (MLMC) method, which improves the accuracy and efficiency of Monte Carlo simulations. MLMC achieves this by leveraging simulations at different levels of discretization of the SDE and efficiently combining the results. This approach significantly reduces computational cost while maintaining the desired level of precision. Afterward, we introduce another approach that establishes a connection between partial differential equations (PDEs) and the expectation of the pay-off. This method takes advantage of the fact that the pricing of options can often be reformulated as a PDE problem. Finally, we present our numerical results and offer interpretations based on our findings.

2 Classical Monte Carlo

2.1 Principle of the method

The aim of Monte Carlo method in our context is to approximate the expectation of a given payoff P which is a functional F of a solution S of some stochastic differential equation. In this project, we work only with pay-off that depends on one single asset S which follows the dynamic :

$$dS_t = b(t, S_t)\, dt + \sigma(t, S_t)\, dW_t$$

where W_t is an arithmetic brownian motion and b, σ are function that can depend on ω.

Given a generator of independant and indentically distributed sample of our payoff we can approximate the expecation based on law of large number. Assuming $F(S_t, t \in [0,T]) \in L_1(\Omega, \mathcal{F}, P)$ we have :

$$\sum_{k=1}^{n} F(S_t^k, t \in [0,T]) \xrightarrow[n\to\infty]{a.s.} E[F(S_t, t \in [0,T])] = E[P]$$

Where S^k are independant samples of the price. This allow us to define an estimator $\overline{Y}_n$ of $E(P)$ by

$$\overline{Y}_n = \frac{1}{n} \sum_{k=1}^{n} F(S_t^k, t \in [0,T])$$

If furthemore $F(S_t, t \in [0,T]) \in L_2(\Omega, \mathcal{F}, P)$ we can construct a confidence interval based on central limit theorem :

$$I_\delta = \left[\overline{Y}_n - z_{\frac{\delta}{2}} \frac{\overline{\sigma}_n}{\sqrt{n}},\ \overline{Y}_n + z_{\frac{\delta}{2}} \frac{\overline{\sigma}_n}{\sqrt{n}} \right]$$

$$with \quad \overline{\sigma}_n^2 = \frac{1}{n-1} \sum_{k=1}^{n} \left(F(S_t^k, t \in [0,T]) - \overline{Y}_n \right)^2$$

with $z_{\frac{\delta}{2}}$ the corresponding quantile of a center gaussian variable so that

$$P(E[P] \in I_\delta) \approx 1 - \delta$$

2.2 Euler scheme

To simulate paths of S_t under the dynamics

$$dS_t = b(t, S_t)\, dt + \sigma(t, S_t)\, dW_t,$$

we introduce a uniform discretization of the time interval $[0,T]$. Let us define the partition $0 = t_0 < t_1 < \cdots < t_m = T$ where $t_i = i \times \Delta t$ and $\Delta t = \frac{T}{m}$. We denote by $\hat{S}_{t_i}$ the approximation of S_{t_i}. The Euler scheme then reads:

$$\hat{S}_{t_{i+1}} = \hat{S}_{t_i} + b\big(t_i, \hat{S}_{t_i}\big)\, \Delta t + \sigma\big(t_i, \hat{S}_{t_i}\big)\, \Delta W_i,$$

where $\Delta W_i = W_{t_{i+1}} - W_{t_i}$ are increments of the Brownian motion. In practice, these increments are sampled as

$$\Delta W_i \sim \mathcal{N}(0, \Delta t).$$

Once a path $\{\hat{S}_{t_i}\}_{0 \le i \le m}$ has been simulated, we compute the payoff as:

$$\hat{P} = F\big(\{\hat{S}_{t_i}\}_{0 \le i \le N}\big).$$

To estimate the option price $E[P]$, we generate n independent paths, yielding the independent samples $\hat{P}_1, \ldots, \hat{P}_n$. We then define:

$$\overline{Y}_n = \frac{1}{n} \sum_{k=1}^{n} \hat{P}_k.$$

Under mild assumptions, $\overline{Y}_n$ converges almost surely to $E[P]$ by the law of large numbers, and we can apply the central limit theorem to build confidence intervals as described in the previous section.

2.3 Error Decomposition

Using the Euler scheme to approximate the diffusion process, the Monte Carlo estimation error $\mathcal{E}$ is given by

$$\mathcal{E} = \frac{1}{n} \sum_{i=1}^{n} F\big(\bar{S}_T^{m,i}\big) \ - \ \mathbb{E}\big[F(S_T)\big]$$

$$= \underbrace{\left(\frac{1}{n} \sum_{i=1}^{n} F\big(\bar{S}_T^{m,i}\big) \;-\; \mathbb{E}\big[F(\bar{S}_T^m)\big] \right)}_{\text{statistical error}} + \underbrace{\big(\mathbb{E}\big[F(\bar{S}_T^m)\big] \;-\; \mathbb{E}\big[F(S_T)\big]\big)}_{\text{bias}}$$

where $\{\bar{S}_T^{m,i}\}_{i\geq 1}$ are independent copies of $\bar{S}_T^m$.

We can identify two main sources of error:

- **Statistical error.**
  The term
  $$\frac{1}{n} \sum_{i=1}^{n} F\big(\bar{S}_T^{m,i}\big) \;-\; \mathbb{E}\big[F(\bar{S}_T^m)\big]$$
  converges on the order of $\frac{1}{\sqrt{n}}$ according to the Central Limit Theorem. Increasing n reduces this part of the error.

- **Bias.**
  The term
  $$\mathbb{E}\big[F(\bar{S}_T^m)\big] \;-\; \mathbb{E}\big[F(S_T)\big]$$
  converges on the order of $\frac{1}{m}$, where m is linked to the time-discretization parameter (e.g., the number of time steps). This convergence rate is given by the weak error (absolute value of the bias here) of the Euler scheme. Increasing m (i.e., refining the discretization) reduces the bias.

3 Multi Level approach

Monte Carlo (MC) methods approximate the expectation of a given payoff by simulating many independent paths with a uniform discretization of the underlying stochastic differential equation (SDE). However, this can require a large number of simulations to achieve a tight confidence interval, particularly when the time step is fine.

In the classical setting, if we have an Euler scheme with step size h and N samples, the mean-square error (MSE) often satisfies

$$\mathrm{MSE}(\widehat{Y}) \;=\; c_1 \frac{1}{N} \;+\; c_2\, h^2.$$

One must choose $h = \mathcal{O}(\varepsilon)$ and $N = \mathcal{O}(\varepsilon^{-2})$ to achieve an MSE of $\mathcal{O}(\varepsilon^2)$. Since each path costs $\mathcal{O}(1/h)$ time steps, the total computational complexity is

$$\mathcal{O}\big(\varepsilon^{-2} \times \varepsilon^{-1}\big) \;=\; \mathcal{O}\big(\varepsilon^{-3}\big).$$

Multi-Level Monte Carlo (MLMC), tackles this cost explosion by using multiple discretization levels (from coarse to fine) and combining them in a telescoping sum. This lets us *reduce variance* without changing the finest step size h_L, so the bias remains the same as in single-level MC but at a significantly lower total cost (often $\mathcal{O}(\varepsilon^{-2}\,(\log \varepsilon^{-1})^2)$ instead of $\mathcal{O}(\varepsilon^{-3})$).

3.1 Telescoping Sum Formulation

Consider approximations $P_0, P_1, \ldots, P_L$ to the true payoff P at increasingly refined time steps Δt_ℓ. We have

$$\mathbb{E}[P_L] \;=\; \mathbb{E}[P_0] \;+\; \sum_{\ell=1}^{L} \Big(\mathbb{E}[P_\ell] - \mathbb{E}[P_{\ell-1}] \Big).$$

Rather than approximating $\mathbb{E}[P_L]$ directly, we independently estimate each correction $\mathbb{E}[P_\ell - P_{\ell-1}]$ and $\mathbb{E}[P_0]$ via Monte Carlo. Namely,

$$\widehat{Y}_\ell \;=\; \frac{1}{n_\ell} \sum_{i=1}^{n_\ell} \Big(P_\ell^{(i)} - P_{\ell-1}^{(i)} \Big),$$

where $P_\ell^{(i)}$ and $P_{\ell-1}^{(i)}$ share the *same Brownian increments* from level $\ell - 1$ to ℓ. Crucially, the sets of paths for $\widehat{Y}_\ell$ and $\widehat{Y}_{\ell-1}$ are distinct, preserving *independence* across levels.

3.2 Variance and Cost Reduction

Observe that $P_\ell - P_{\ell-1}$ often has *much lower variance* than P_ℓ alone, thanks to the correlation of paths at levels ℓ and $\ell - 1$. This allows using fewer samples n_ℓ at finer levels. The cost is

$$\text{Cost} \;=\; \sum_{\ell=0}^{L} \big(N_\ell \times h_l^{-1} \big),$$

and the variance of the MLMC estimator is

$$\text{Var}\Big(\sum_{\ell=0}^{L} \widehat{Y}_\ell \Big) \;=\; \sum_{\ell=0}^{L} \frac{V_\ell}{n_\ell},$$

where $V_\ell = \text{Var}(P_\ell^{(i)} - P_{\ell-1}^{(i)})$.

By balancing the number of samples n_ℓ across levels (e.g. $N_\ell \propto \sqrt{V_\ell \, \Delta t_\ell}$), one obtains large savings compared to single-level MC.

3.3 Weak Error, Bias, and Computational Complexity

Let $h_\ell = T/M^\ell$. Under reasonable assumptions (Euler scheme, Lipschitz payoff, etc.), the *weak error* satisfies

$$\big| \mathbb{E}[P_L] - \mathbb{E}[P] \big| \;=\; \mathcal{O}(h_L), \quad \text{Var}(P_\ell - P) \;=\; \mathcal{O}(h_\ell).$$

Hence,

$$\text{Var}(P_\ell - P_{\ell-1}) \;\le\; \big(\text{Var}(P_\ell - P)^{1/2} + \text{Var}(P_{\ell-1} - P)^{1/2} \big)^2 \;=\; \mathcal{O}(h_\ell).$$

One can show that choosing $L \approx \log(\varepsilon^{-1}) / \log(M)$ ensures $h_L \approx \varepsilon$, so the bias is $\mathcal{O}(\varepsilon)$. Meanwhile, a suitable distribution of samples n_ℓ yields total cost

$$\text{Cost} \;=\; \mathcal{O}\big(\varepsilon^{-2} (\log \varepsilon^{-1})^2 \big),$$

rather than $\mathcal{O}(\varepsilon^{-3})$. Thus, MLMC retains the same bias order but achieves a substantially lower variance–cost product than single-level MC.

MLMC significantly reduces the number of expensive, high-resolution simulations by supplementing them with cheaper, coarser simulations that correct most of the error. The resulting complexity for an MSE $\mathcal{O}(\varepsilon^2)$ is often $\mathcal{O}(\varepsilon^{-2}(\log \varepsilon^{-1})^2)$, a notable improvement over the $\mathcal{O}(\varepsilon^{-3})$ cost of classical single-level MC.

3.4 Bias estimation and stopping criterion

For a Lipschitz payoff and an Euler discretization, as $l \to \infty$, the leading-order bias scales like

$$\mathbb{E}\big[P - \widehat{P}_l\big] \;\approx\; c_1\, h_l.$$

In addition, one often observes that

$$\mathbb{E}\big[\widehat{P}_l - \widehat{P}_{l-1}\big] \;\approx\; (M-1)\,\mathbb{E}\big[P - \widehat{P}_l\big],$$

where M is the ratio between successive levels $h_l = T/M^l$. Hence, *without knowing the true payoff* P, we can still *estimate* the remaining bias in $\widehat{P}_l$ by examining $\widehat{P}_l - \widehat{P}_{l-1}$.

Stopping criterion. In practice, we increase L until the magnitude of this estimated bias is below a desired tolerance, say $\frac{\varepsilon}{\sqrt{2}}$. In the referenced article, they enforce

$$\max\Big\{\, M^{-1}\,\big|\widehat{Y}_{L-1}\big|,\, \big|\widehat{Y}_L\big|\Big\} \;<\; \frac{1}{\sqrt{2}}\,(M-1)\,\varepsilon,$$

ensuring that convergence is not "accidental" (i.e. not just a chance cancellation). Here, $\widehat{Y}_L$ typically denotes the estimate of $\mathbb{E}[\widehat{P}_L - \widehat{P}_{L-1}]$.

Richardson extrapolation. Another approach is to *eliminate* the leading-order bias by leveraging the relation

$$\mathbb{E}\big[\widehat{P}_l - \widehat{P}_{l-1}\big] \;\approx\; (M-1)\,\mathbb{E}\big[P - \widehat{P}_l\big].$$

Hence, we can *add back* the term $(M-1)^{-1}\big(\widehat{P}_l - \widehat{P}_{l-1}\big)$ to the estimator $\widehat{Y}$ to offset the dominant bias. Concretely, one can write a *combined* estimator like

$$\frac{M}{M-1}\Big\{\widehat{Y}_0 \;+\; \sum_{\ell=1}^{L}\Big(\widehat{Y}_\ell \;-\; M^{-1}\,\widehat{Y}_{\ell-1}\Big)\Big\},$$

which implements a Richardson correction. Using this technique, the stopping criterion might be

$$\big|\widehat{Y}_L \;-\; M^{-1}\,\widehat{Y}_{L-1}\big| \;<\; \frac{1}{\sqrt{2}}\,(\,M^2-1\,)\,\varepsilon,$$

ensuring the *leading-order* term in the bias is removed and the remaining (higher-order) bias is below the prescribed threshold.

3.5 MLMC Algorithm

We summarize the Multi-Level Monte Carlo procedure as follows:

1. **Initialization:** Set the coarsest level $L = 0$. Choose an initial number of samples $N_L = 10^4$ at this level and estimate the single-sample variance V_L.

2. **Compute optimal sample sizes:** For each level $\ell = 0, \ldots, L$, define

$$N_\ell \;=\; \left\lceil 2\,\varepsilon^{-2}\,\sqrt{V_\ell\,h_\ell}\,\Big(\sum_{j=0}^{L}\sqrt{\frac{V_j}{h_j}}\,\Big)\right\rceil,$$

   where h_ℓ is the time step at level ℓ and V_ℓ is the estimated single-sample variance of $\widehat{P}_\ell^{(i)} - \widehat{P}_{\ell-1}^{(i)}$.

3. **Update samples:** At each level ℓ, if the newly prescribed N_ℓ exceeds the current number of samples, draw additional simulations so that the total is N_ℓ. Recompute (or update) the estimates $\widehat{Y}_\ell$ of $\mathbb{E}[\,P_\ell - P_{\ell-1}\,]$ accordingly.

4. **Check stopping criterion:** If $L \geq 2$, test for convergence using a bias estimate such as

$$\max\Big\{M^{-1}\,\big|\widehat{Y}_{L-1}\big|,\,\big|\widehat{Y}_L\big|\Big\} \;<\; \frac{1}{\sqrt{2}}\,(\,M-1\,)\,\varepsilon,$$

   or a Richardson extrapolation version

$$\big|\widehat{Y}_L - M^{-1}\,\widehat{Y}_{L-1}\big| \;<\; \frac{1}{\sqrt{2}}\,(\,M^2-1\,)\,\varepsilon.$$

   If this condition is satisfied, the bias is below $\varepsilon/\sqrt{2}$. Combined with the variance choice (Step 2) ensuring $\mathrm{Var}\big(\widehat{P}_{\mathrm{MLMC}}\big) \leq \frac{1}{2}\,\varepsilon^2$, the total MSE stays below ε^2.

5. **Increase level if necessary:** If $L < 2$ or the stopping criterion is *not* met, increment $L \leftarrow L+1$, set an initial $N_L = 10^4$ at the new level, and return to Step 2.

This choice of sample sizes N_ℓ ensures that the variance contribution from each level is controlled so that $\sum_{\ell=0}^{L}\mathrm{Var}(\widehat{Y}_\ell) \leq \frac{1}{2}\,\varepsilon^2$, while the stopping test for L ensures the bias is below $\frac{1}{\sqrt{2}}\,\varepsilon$. Together, they guarantee that the overall mean-square error is at most ε^2.

4 Finite difference approach

4.1 Finite Difference Methods for the Black–Scholes PDE

We consider the Black–Scholes PDE for a function $u = u(S,t)$:

$$\frac{\partial u}{\partial t} \;+\; r\,S\,\frac{\partial u}{\partial S} \;+\; \frac{1}{2}\,\sigma^2\,S^2\,\frac{\partial^2 u}{\partial S^2} \;-\; r\,u \;=\; 0,$$

where r is the risk-free rate and σ is the volatility. This PDE is *linear* and *parabolic* in nature.

We can rewrite the PDE as

$$\partial_t u + Au + f = 0 \quad \text{in } [0,T) \times (0,\beta),$$

where

$$Au \;=\; \tfrac{1}{2}\,\sigma^2\,S^2\,\frac{\partial^2 u}{\partial x^2} \;+\; r\,S\,\frac{\partial u}{\partial x}, \quad f = -\,r\,u.$$

We impose the terminal condition $u(T,\cdot) = g$ in (α,β), and Dirichlet boundary conditions

$$u(\cdot,0) = 0 \quad \text{and} \quad u(\cdot,\beta) = S - K\,e^{-r\,T}$$

for a European call option.

We discretize the domain in S with step ΔS and in t with step Δt. Hence, the grid mesh is

$$\{(t_n, x_i)\} \;=\; \big(n\,h,\; i\,\delta\big), \quad h = \tfrac{T}{m},\;\; \delta = \tfrac{\beta}{l+1}.$$

Denoting the approximate solution at $\big(S_i, t_j\big)$ by $u_{i,j}$, we replace partial derivatives by finite differences.

Approximating the Spatial Derivatives

Assuming $\sigma \neq 0$, we approximate spatial derivatives by central differences. For instance,

$$A\,u(t_n,x_i) \;\approx\; \left(\frac{\sigma^2\,S_i^2}{2\,\delta^2} \;-\; \frac{r\,S_i}{2\,\delta}\right) u_{i-1}^n \;-\; \frac{\sigma^2\,S_i^2}{\delta^2}\,u_i^n \;+\; \left(\frac{\sigma^2\,S_i^2}{2\,\delta^2} \;+\; \frac{r\,S_i}{2\,\delta}\right) u_{i+1}^n.$$

Approximating the Time Derivative

We can use three main schemes for $\frac{\partial u}{\partial t}$:

- Explicit scheme (forward in time)
- Implicit scheme (backward in time)
- Crank–Nicolson scheme (central in time)

The θ-Scheme

A unified approach (the so-called θ-scheme) is:

$$u_n \;=\; \big(I - \theta\,h\,A\big)^{-1}\Big[\big(I + (1-\theta)\,h\,A\big)\,u_{n+1}\Big] \;+\; h\,f_\Delta,$$

where A is the tridiagonal matrix encoding the spatial discretization, and θ controls the scheme:

$$\theta = 0 \quad \text{(explicit)}, \qquad \theta = 1 \quad \text{(implicit)}, \qquad \theta = 0.5 \quad \text{(Crank–Nicolson)}.$$

4.2 Convergence and Stability of the θ-Scheme for the Black–Scholes PDE

Suppose that for all $x \in \Omega$,

$$|b(x)|\,\delta \;\le\; \sigma^2(x), \quad \text{and} \quad \sigma^2(x)\,\big(1-\theta\big)\,h \;\le\; \delta^2.$$

Under these conditions, the θ-scheme is **stable**. In particular, the second condition is trivially satisfied when $\theta = 1$ (fully implicit scheme), and if σ is uniformly elliptic, then for sufficiently small δ the first condition holds as well.

By combining stability with consistency, one obtains convergence. In more detail:

- If there is a sufficiently smooth exact solution $u(x,t)$ to the PDE,
- and if the conditions above hold (ensuring stability),

then the θ-scheme converges. Specifically, one can show that

$$\max_{n,i} \big\| \, u_i^n - u\big(t_n, x_i\big) \big\|_\infty \;=\; \begin{cases} \mathcal{O}\big(h+\delta^2\big), & \theta \neq \frac{1}{2}, \\ \mathcal{O}\big(h^2+\delta^2\big), & \theta = \frac{1}{2} \text{ (Crank–Nicolson)}, \end{cases}$$

where u_i^n is the numerical solution at time-step t_n and grid-point x_i.

Remark on Explicit Scheme ($\theta = 0$). If $\theta = 0$ (purely explicit), we see from $\sigma^2(x)\,(1-\theta)\,h \le \delta^2$ that $\sigma^2(x)\,h \le \delta^2$. For small δ, this condition can be violated easily unless we refine sufficiently in time (i.e. make N large) so that h is also small. This shows why the explicit scheme can be unstable if the time step is too large (N too small) relative to the spatial discretization δ. Hence, for many practical problems, $\theta = 1$ (fully implicit) or $\theta = \frac{1}{2}$ (Crank–Nicolson) are preferred for stability reasons.

5 Applications

5.1 Option Pay-off

In this section, we illustrate four types of pay-offs that can be used in our Monte Carlo framework and how to approximate it with Euler sheme. We suppose that we have a constant interest rate r on the market.

European Option. A standard European call (resp. put) with strike K can be written as

$$P_{\text{euro}} = e^{-rT}\,\big(\hat{S}_T - K\big)_+ \quad (\text{resp. } e^{-rT}\,\big(K - \hat{S}_T\big)_+),$$

where $\hat{S}_T$ is the simulated asset price at maturity T.

Asian Option. For the Asian call option (resp. put), the payoff depends on the average of the underlying price over $[0, T]$.

$$\frac{1}{T}\int_0^T S_t\, dt$$

We approximate the time integral by a trapezoidal rule:

$$\bar{S}_m = \sum_{n=1}^{m} \tfrac{1}{2}\left(\hat{S}_n + \hat{S}_{n-1}\right) h_m,$$

where $h_m = \frac{T}{m}$ and $\hat{S}_n$ denotes the approximation of Euler sheme of the price at time step n. The Asian payoff then reads

$$P_{\text{asian}} = e^{-rT}\left(\bar{S}_m - K\right)_+, \quad (\text{resp. } e^{-rT}\left(K - \bar{S}_m\right)_+)$$

Lookback Option. A lookback call payoff at maturity T is of the form

$$P_{\text{lookback}} = e^{-rT}\left(\hat{S}_T - \min_{0\le t\le T} \hat{S}_t\right).$$

In practice with a geometric brownian dynamic for the price we approximate the minimum by

$$\hat{S}_{\min,m} = \left(\min_m \hat{S}_m\right)\left(1 - \beta^* \sigma \sqrt{h_m}\right),$$

where $\beta^* \approx 0.5826$ is a correction factor, σ is the volatility parameter in the price dynamics and $h_m = \frac{T}{m}$. The payoff then becomes

$$P_{\text{lookback}} = e^{-rT}\left(\hat{S}_T - \hat{S}_{\min,\ell}\right).$$

Digital Option. A digital call option (resp. put) pays 1 unit of cash if $\hat{S}_T$ exceeds the strike K at maturity, and 0 otherwise. Its payoff is

$$P_{\text{digital}} = e^{-rT}\,\mathbf{1}_{\{\hat{S}_T > K\}}, \quad (\text{resp. } e^{-rT}\,\mathbf{1}_{\{\hat{S}_T < K\}})$$

5.2 Multi Level numerical result

A commonly used model in financial mathematics assumes that the asset price S_t follows a Geometric Brownian Motion. Under the risk neutral measure (associated with the numeraire induced by the interest rate r just above), the stochastic differential equation reads:

$$dS_t = rS_t\, dt + \sigma\, S_t\, dW_t, \quad S_0 \text{ given},$$

where r is the constant interest rate on the market, $\sigma > 0$ is the volatility, and W_t denotes a standard Brownian motion.

We have replicated all results from the original paper of the authors using the exact same number of samples, parameters, and methods. The graphs are organized as follows:

Variance plot (top left):
In this plot, we are interested in the evolution of the variance of the option prices P_l and the differences in prices $P_l - P_{l-1}$ at different refinement levels. At level l, the price paths S_l are discretized using M^l steps in the Euler scheme. We observe that for all options, $V(P_l)$ is an increasing function of l. This suggests that for classical Monte Carlo methods, reducing the bias by using a finer Euler scheme simultaneously increases the statistical error, requiring more samples.
As expected, $V(P_l - P_{l-1})$ is a decreasing function of l, as $M^l - M^{l-1}$ becomes very small. This highlights the advantage of multilevel methods, which require fewer and fewer samples to achieve a similar variance in the estimator as l increases. While classical Monte Carlo methods distribute effort uniformly to approximate P_L, multilevel methods allocate more effort to coarser levels of $P_l - P_{l-1}$.

Absolute mean plot (top right):
Here, we are interested in $|E(P_l)|$, $|E(P_l - P_{l-1})|$ (naive stopping criterion), and $|E((Y_l - Y_{l-1})/M)|$ (Richardson extrapolation stopping criterion), where $Y_l = P_l - P_{l-1}$. This allows us to compare the stopping criteria of the algorithm, which characterize the weak error as

$$E(\hat{P}_l - \hat{P}_{l-1}) \approx (M-1)E(P - \hat{P}_l).$$

When one stopping criterion is smaller than another, it suggests that the algorithm will stop at a smaller or equal refinement level l, reducing the time complexity needed to achieve the desired RMSE. As expected, we observe that $|E(P_l - P_{l-1})|$ and $|E((Y_l - Y_{l-1})/M)|$ are decreasing functions of l, ensuring that the algorithm will stop. Additionally, Richardson's stopping criterion is smaller in most cases, indicating greater effectiveness.
Another interpretation of this plot is the convergence of the weak error. In most cases, the slopes are parallel, suggesting that the weak error is inversely proportional to M^l and, hence, proportional to $h_l = M^{-l}$.

Number of samples stopping criterion comparison plot (bottom left):
In this plot, we are interested in the evolution of the number of samples at each level for a given aimed RMSE (ϵ). We compare the number of samples when using Richardson extrapolation versus not using it. What we can expect is that when the Richardson stopping criterion is significantly smaller than the naive one (as observed in the top-right plot), there will be a difference in the number of levels used during the algorithm. This is indeed what we observe. Furthermore, in such cases, the number of samples used at each level with Richardson extrapolation is smaller, as the optimal number of samples is proportional to the sum of the variances at each level:

$$N_l = \left\lceil 2\epsilon^{-2}\sqrt{V_l h_l}\left(\sum_{l=0}^{L}\sqrt{\frac{V_l}{h_l}}\right)\right\rceil, \quad V_l = V(\hat{Y}_l)N_l.$$

Therefore, when we add a level, we mechanically increase the number of samples. This is why Richardson extrapolation is, in most cases, highly effective and has a significant impact on time complexity, as we will observe in the last plot type.

Classic and Multi-level complexity comparison (bottom right):
Here, we compare the time complexity of classic and multi-level methods, with or without Richardson extrapolation. To apply the same termination criterion, we run the classic approach alongside

the multi-level algorithm. In multi-level methods, to achieve the same variance at each level and ensure a statistical error $< \frac{1}{\epsilon^2}$, we use $N_l^* = 2\epsilon^{-2}V(P_l)$. The final estimator is the mean over all levels.

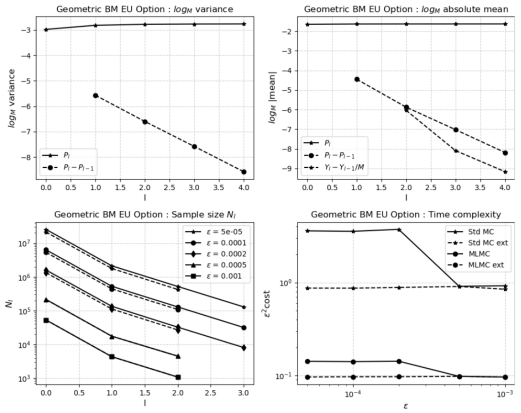
We define the time complexity of the multi-level algorithm as the total number of samples weighted by the inverse of the time steps at levels l and $l-1$:

$$C = N_0 + \sum_{l=1}^{L} N_l \left(M^l + M^{l-1}\right).$$

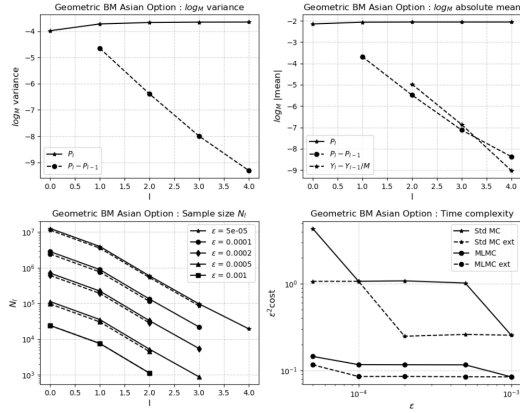
For the classic method, we similarly define time complexity as the total number of samples weighted by the inverse of the time steps at level l:

$$C^* = \sum_{l=0}^{L} N_l M^l.$$

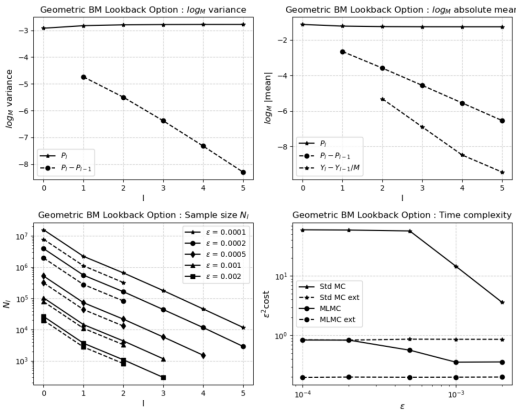
We plot the MSE ϵ against the time complexity scaled by the squared MSE, $\epsilon^2 C$. As expected, we observe that in most cases, the multi-level method and Richardson extrapolation achieve a smaller time complexity compared to the classic method and the naive stopping criterion.



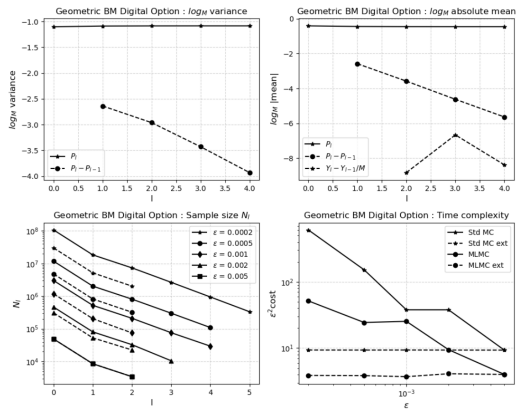
(a) European option numerical result



(b) Asian option numerical result



(c) Lookback option numerical result



(d) Digital option numerical result

Figure 1: Geometric Brownian motion numerical results for various options

The Heston stochastic volatility model describes the joint dynamics of an asset price S_t and its variance V_t. Under the risk-neutral measure, the model is given by

$$dS_t = r\, S_t\, dt \;+\; \sqrt{V_t}\, S_t\, dW_t^1,$$

$$dV_t = \lambda\big(\theta - V_t\big)\, dt \;+\; \xi\, \sqrt{V_t}\, dW_t^2,$$

with initial conditions $S_0 > 0$, $V_0 > 0$. Here:

- r is the constant risk-free interest rate.
- σ is the long-term mean.
- λ is the rate at which V_t reverts to θ.
- ξ is the volatility-of-volatility parameter.

- W_t^1 and W_t^2 are Brownian motions with correlation ρ, i.e., $\mathrm{d}\langle W^1, W^2\rangle_t = \rho\,\mathrm{d}t$.

For a uniform time grid $0 = t_0 < t_1 < \cdots < t_m = T$ with $\Delta t = \frac{T}{m}$, an Euler-type scheme for $\{S_{t_i}, V_{t_i}\}$ can be written as:

$$\hat{V}_{t_{i+1}} = \hat{V}_{t_i} \;+\; \lambda\big(\theta - \hat{V}_{t_i}\big)\,\Delta t \;+\; \xi\,\sqrt{\max\{\hat{V}_{t_i},\,0\}}\;\Delta W_i^2,$$

$$\hat{S}_{t_{i+1}} = \hat{S}_{t_i} \;+\; r\,\hat{S}_{t_i}\,\Delta t \;+\; \sqrt{\max\{\hat{V}_{t_i},\,0\}}\;\hat{S}_{t_i}\;\Delta W_i^1,$$

where

$$\Delta W_i^1 = \sqrt{\Delta t}\;Z_i^1, \quad \Delta W_i^2 = \rho\,\sqrt{\Delta t}\;Z_i^1 \;+\; \sqrt{1-\rho^2}\,\sqrt{\Delta t}\;Z_i^2,$$

and $Z_i^1, Z_i^2 \sim \mathcal{N}(0,1)$ are independent standard normal variables. In practice, one ensures that $\hat{V}_{t_i}$ remains nonnegative by taking $\max\{\hat{V}_{t_i},\,0\}$ in the square root.

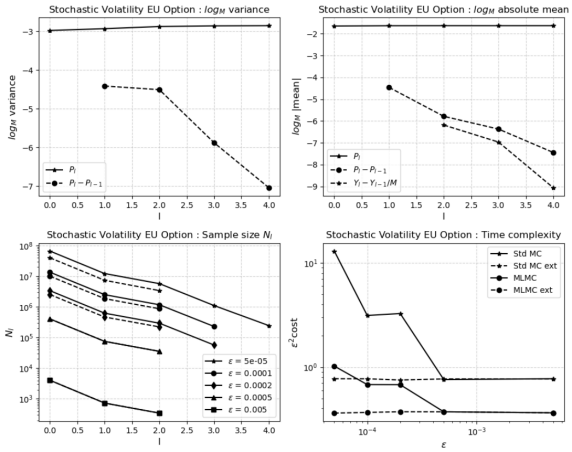


Figure 2: Heston model european option numerical result

5.3 Finite difference numerical result

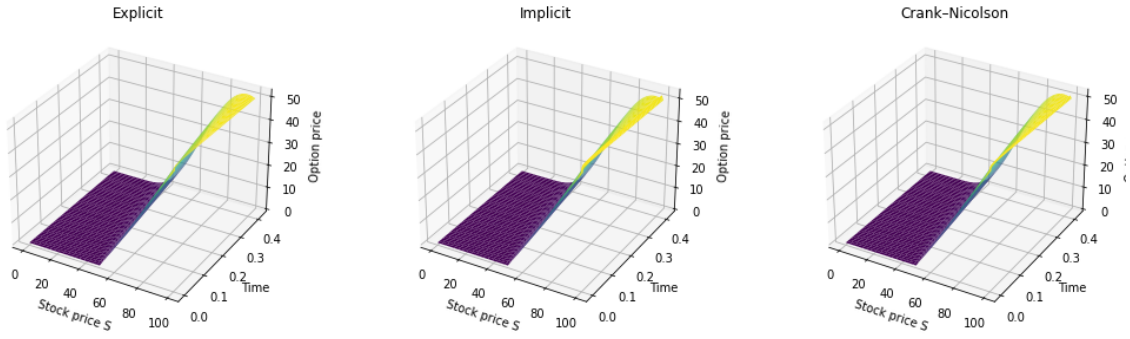


Figure 3: Option Value surfaces for the three schemes with N = 10000

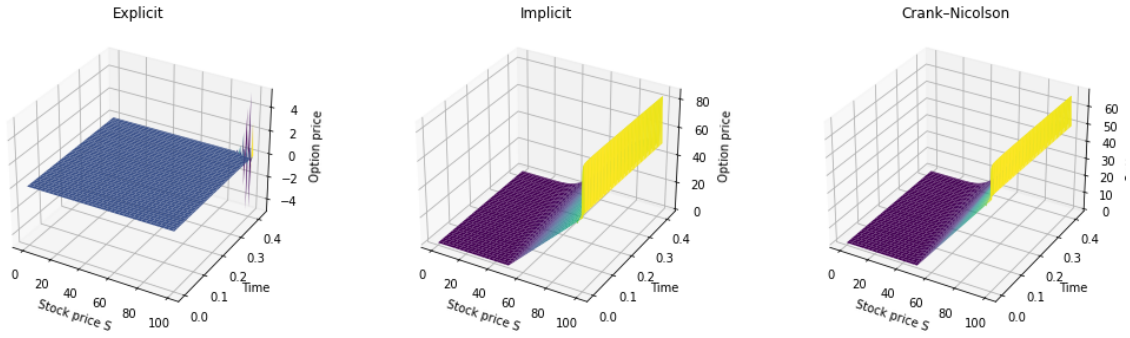


Figure 4: Option Value surfaces for the three schemes with $N = 500$.
Note: As discussed earlier, the explicit scheme can become unstable if N is too small.

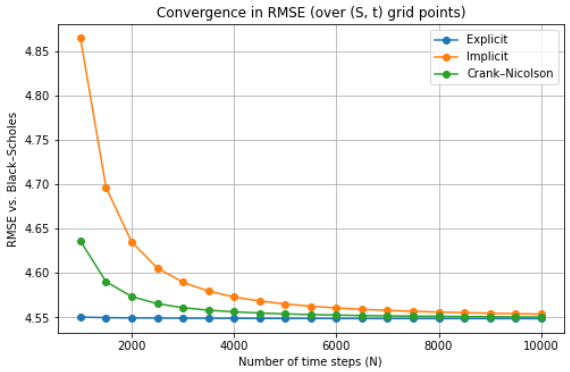


Figure 5: Option Value surfaces for the three schemes with N = 500

6 Conclusion

In this work, we have shown how the Multi-Level Monte Carlo (MLMC) method can drastically reduce the computational cost of estimating expectations as compared to classical Monte Carlo. By exploiting a telescoping sum across a hierarchy of discretizations, MLMC balances the number of samples required at each level, matching the variance reduction to the cost of simulating finer (and more expensive) discretizations.

In particular, MLMC preserves the same order of bias associated with the finest discretization as standard Monte Carlo, while significantly reducing the variance contribution of the finer levels through correlated simulations. Consequently, the overall complexity to achieve a mean-square error of $\mathcal{O}(\varepsilon^2)$ can often be lowered from $\mathcal{O}(\varepsilon^{-3})$ to $\mathcal{O}\big(\varepsilon^{-2}(\log \varepsilon)^2\big)$, or better, depending on the problem specifics and the discretization scheme.

These efficiency gains make MLMC an attractive choice in scenarios where traditional Monte Carlo methods become prohibitively expensive. The framework is also flexible: it can be extended to various stochastic models, payoff structures, or even to different numerical schemes beyond basic Euler time stepping.